

VŠB – Technická Univerzita Ostrava
Fakulta elektrotechniky a informatiky

Bakalárska práca

VŠB – Technická Univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

**Nástroj pre tvorbu, kompletáciu a evaluáciu štruktúrovaných
dokumentov**

**Tool for design, completion and evaluation of structured
documents**

Zadání bakalářské práce

Student:

Marián Miko

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Nástroj pro tvorbu, kompletaci a evaluaci strukturovaných dokumentů

Tool for Design, Completion and Evaluation of Structured Documents

Zásady pro vypracování:

1. Seznamte se s nástroji pro tvorbu strukturovaných dokumentů. Zaměřte se na problematiku nástrojů pro podporu rozhodování v oblasti obchodního plánování a budování prodejních a marketingových strategií.
2. Proveďte analýzu a návrh implementace nástroje, který bude schopen definovat struktury dokumentů, šablony dokumentů, průvodce pro vyplňování dokumentů, validátor obsahu, výpočetní moduly pro dokumenty a vizualizaci výsledků.
3. Cílem práce je provést implementaci tohoto řešení jako desktopové aplikace tak i aplikace s možností kooperativní spolupráce v prostředí internetu.
4. Proveďte ověření v praxi. Výstupem bude nástroj použitelný jak ve výuce tak jako podpůrný prostředek pro širokou veřejnost.
5. Výslednou aplikaci porovnejte s existujícími nástroji.

Seznam doporučené odborné literatury:

- [1] Business Plan Pro, Sales & Marketing (www.paloalto.com)
- [2] Mária Režňáková, Vojtěch Koráb, Jiří Peterka, Podnikatelský plán, CPress, s. 216, 2007, EAN: 9788025116050
- [3] Greg W. Marshall, Michael R. Solomon, Elnora W. Stuart, Marketing, CPress, s.512, 2006, EAN: 9788025112731
- [4] Hana Klčová, Petr Sodomka, Informační systémy v podnikové praxi, CPress, s. 504, 2011, EAN: 9788025128787

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Radoslav Fasuga, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Čestné vyhlásenie

Čestne vyhlasujem, že záverečnú prácu som vypracoval samostatne a že som uviedol všetku použitú literatúru.

Dátum:


.....
(podpis študenta)

ABSTRAKT

Naša bakalárska práca sa zaoberá analýzou a implementáciou nástroja, určeného na tvorbu a následne používanie elektronických formulárov. Prvá časť práce sa venuje už existujúcim riešeniam na trhu a približuje ich funkčnosť, kde následne v druhej časti, stanoví jasné ciele ako budeme pri implementácii postupovať. Nami navrhnuté riešenie implementujeme pomocou stanovenia formátu v akom budú dáta ukladané a vytvorenia modulu schopného samostatnej práce nad ním. V ďalšej kapitole. Posledná kapitola sa venuje zostaveniu našej práce do konečného výsledku. Finálny produkt tejto práce je schopný generovať elektronické formuláre a pracovať s nimi.

ABSTRACT

This bachelor's thesis describes the analysis and implementation of a tool designed that is intended to use to work with electronic forms. The first chapter of this work addresses the already existing tools for this solution on the market and point out their functionality, where we will later in the second part establishes goals how will we move on. In the next chapter, the from us designed solution will be implemented by the format in witch the data will be stored and the creation of a module, capable to work on it. The last chapter addresses the assembly of our work in to the final result. The final product of this work will be able to generate electronic forms and work with them.

Kľúčové slová: XML, XPath, formulár, element, javascript, MySQL

Keywords: XML, XPath, formulár, element, javascript, MySQL

Podakovanie

Týmto by som chcel poďakovať vedúcemu bakalárskej práce, Ing. Radoslavovi Fasugovi, Ph.D., za pomoc pri písaní tejto bakalárskej práce, rodine, priateľom a tiež všetkým, ktorí mi pomohli užitočnými radami a cennými myšlienkami.

OBSAH

| | |
|---|----|
| Úvod..... | 9 |
| 1 Formuláre | 10 |
| 1.1 Elektronický formulár | 10 |
| 1.1.1 Tvorca e-formuláru..... | 10 |
| 1.1.2 Vypĺňače e-formuláru..... | 11 |
| 1.1.3 E – formulárový procesor..... | 11 |
| 1.2 Výhody elektronizácie..... | 11 |
| 1.3 Existujúce systémy pre tvorbu elektronických formulárov | 11 |
| 1.3.1 Zahraničné spoločnosti..... | 11 |
| 1.3.2 Tuzemské spoločnosti | 14 |
| 2 Špecifikácia zadania..... | 16 |
| 2.1 Základný popis a ciele..... | 16 |
| 2.2 Kto bude s IS pracovať..... | 17 |
| 2.3 Zoznam vstupov a výstupov systému..... | 18 |
| 2.3.1 Vstupy systému | 18 |
| 2.3.2 Výstupy | 18 |
| 2.4 ERD dieagram | 19 |
| 2.5 Návrh užívateľského rozhrania | 20 |
| 2.6 Použité technológie | 22 |
| 2.6.1 Serverové jadro..... | 22 |
| 2.6.2 Použitá databáza | 22 |
| 2.6.3 Ajax, JQuery a javascript | 22 |
| 3 Formát a modul AIDoX | 23 |
| 3.1 XML..... | 23 |
| 3.2 Jazyk XPath..... | 24 |
| 3.3 Formát AIDoX | 25 |

| | | |
|-------|--------------------------------------|----|
| 3.3.1 | Model..... | 25 |
| 3.3.2 | Controller..... | 25 |
| 3.3.3 | View | 26 |
| 3.4 | AIDoX modul..... | 26 |
| 3.4.1 | AIDoX.wrapper..... | 27 |
| 3.4.2 | AIDoX.xml..... | 28 |
| 3.4.3 | AIDoX.format | 28 |
| 3.4.4 | AIDoX.evaluate..... | 29 |
| 3.5 | AIDoX_visual.js..... | 30 |
| 4 | Implementácia | 31 |
| 4.1 | Použité externé nástroje..... | 31 |
| 4.2 | Aplikácia na strane užívateľa | 32 |
| 4.3 | Aplikácia na strane servera..... | 33 |
| 4.3.1 | Model..... | 33 |
| 4.3.2 | Controller..... | 34 |
| 4.3.3 | View | 34 |
| 4.3.4 | Library..... | 34 |
| 4.4 | Overovanie prihlásenia a práva | 34 |
| | Záver..... | 36 |

Úvod

Neustáli vývoj informačných technológií spôsobuje, že množstvo aplikácií, ktoré bývali neoddeliteľnou súčasťou operačných systémov, nachádzajú postupne oveľa flexibilnejšie alternatívy v prostredí internetu. Užívateľ nie je viac nútený inštalovať systém na svoje zariadenie a sťahovať neustále aktualizácie, ale prístupom všetkých užívateľov na jeden zdroj sú zmeny premietnuté okamžite.

Z tohto dôvodu sme sa rozhodli vypracovať prácu ako webovú aplikáciu. Cieľom práce je analýza a implementácia nástroja schopného práce s elektronickými formulármi. Prácu s formulármi musíme chápať na dvoch úrovniach. Ako nástroj, ktorý dokáže vytvoriť formulár a nadviazať na výpočtovú logiku, tak aj nástroj schopný s týmto výsledným formulárom pracovať. Aby to bolo možné je však potrebné stanoviť formát výsledného dokumentu. Zadanie špecifikuje využitie systému pre edukačné účely a preto sme pridali malý podsystem na správu kurzov, ktorého pre nás dôležitá časť je priradzovanie a vyplňanie našich dokumentov.

Úvod práce sa v prvej kapitole venuje významu elektronických formulárov a približuje ich fungovanie porovnáva už existujúce nástroje zaoberajúce sa touto problematikou.

V druhej kapitole špecifikujeme zadanie a navrhujeme si implementáciu riešenia ako aj definujeme užívateľov systému podľa role v akej budú vystupovať.

Tretia kapitola má za úlohu špecifikáciu formátu v akom budú formuláre ukladané. Popisujú sa tu jeho jednotlivé vrstvy a ich význam. Rovnako je priblížený aj jazyk XPath, ktorý je dôležitou súčasťou systému.

Štvrtá kapitola popisuje implementáciu riešenia ako na strane užívateľa tak aj serverovej časti. Vysvetľujú sa tu základné prvky fungovania a popisujeme tu aj použité nástroje tretích strán.

1 Formuláre

Význam formulárov je v praxi veľký. Na základe nimi získaných informácií z nich dokážu spoločnosti alebo jednotlivci vytvárať výsledné správy alebo získať informácie o klientoch. Správa papierových formulárov je však náročná. Po distribúcií kópie sa môžu niektoré stratiť alebo dokonca zničiť. V konečnom dôsledku je potrebné informácie ručne nahadzovať do systému a môže sa stať, že nie sú ani len čitateľné. Ich najväčším problémom však je že ich treba fyzicky uchovávať a vznikajú rozsiahle kartotéky, v ktorých vyhľadávanie je veľmi obtiažné.

Za formulár však môžeme označiť tlačivo v analógovej alebo neštruktúrovanej elektronickej forme reprezentujúce dané tlačivá.[1]

Predstavujú:

- tlačivá v papierovej podobe
- textové súbory umožňujúce digitálne zverejnenie
- štrukturované dokumenty nespĺňajúce náležitosti elektronického formulára

1.1 Elektronický formulár

Elektronický formulár (ďalej len „e-formulár“) je digitálnou alternatívou ku klasickým papierovým formulárom. Sú užitočným nástrojom na redukciu tlačenia, ukladania a distribúcie množstva predtlačených formulárov. Z dôvodu, že logika asociovaná s formulárom pomáha pri vyplnení, validácii a kalkulácii vstupných údajov od užívateľa jeho vyplňanie je rýchlejšie a efektívnejšie.

Systémy zabezpečujúce tvorbu a vyplňanie e-formulárov by mali obsahovať tri základné komponenty, ktorými sú:

- tvorca e-formuláru
- vyplňáč e-formuláru
- e-formulárový procesor

1.1.1 Tvorca e-formuláru

Pred začatím používania formulárov je potrebné definovať ich štruktúru v XML formáte, vo WYSIWYG režime, vytvorením vstupných sekcií ako sú textové polia, textové plochy, rozbaľovacie

zoznamy a iné vstupné elementy. Jednou z úloh tvorcu formulárov je taktiež asociovanie logiky na jednotlivé elementy, na základe ktorých bude dokument neskôr vyhodnocovaný.

1.1.2 Vypĺňače e-formuláru

Je nástroj pre koncového užívateľa slúžiaci na vypĺňanie formuláru s automatickou kontrolou vstupných údajov, neumožňujúc zadávanie nesprávnych hodnôt, výpočtami závislých elementov a poskytovania nápovedných textov na jednoduchšie pochopenie požiadavok. Validný dokument je možné exportovať do podporovaných formátov (napr. PDF).

1.1.3 E – formulárový procesor

Na zabezpečenie správneho fungovania formuláru je potrebný medzičlánok vo forme formulárového procesoru, ktorý zabezpečuje kontrolu hodnôt podľa stanovenej business logiky.

1.2 Výhody elektronizácie

- vždy je k dispozícii najnovšia verzia formulára a staršie verzie formulárov je možné previesť na aktuálnu verziu [2]
- zabezpečuje konečnú kontrolu bez možnosti zadávania neplatných údajov
- poskytujú možnosť exportu do osobného počítača v podporovanom formáte
- vzhľadom na súčasnú enviromentálnu politiku Európskej únie znižuje spotrebu papiera s čím priamo súvisí aj nižšie administratívne zaťaženie

1.3 Existujúce systémy pre tvorbu elektronických formulárov

Od prechodu z klasických papierových formulárov do elektronickej podoby sa na trhu etablovalo niekoľko, prevažne zahraničných, firiem. Na českom trhu sú využívané prevažne systémy spoločnosti Software602. Konkrétne sa jedná o produkty FormDesigner a FormFiller. V zahraničí sú to software spoločnosti FormDocs, FormMax alebo produkty od spoločnosti Formatta.

1.3.1 Zahraničné spoločnosti

Spoločnosti poskytujúce software podnikom po celom svete zameriavajúcich sa prevažne na veľké podniky.

FormDocs

FormDocs LLC vyvíja software pre tvorbu, vyplňanie a autorizáciu elektronických formulárov už od roku 2003. Firma mala za cieľ uľahčovať zákazníkom prácu s formulármi. Systémy spoločnosti využíva viac ako 100 tisíc malých podnikov a väčšími korporáciami, ako sú Walt Disney, Public Television, Cambridge University Press a British Airways. Táto spoločnosť je vďaka svojím produktom FormDocs a form filling vedúcou spoločnosťou na svete.[4]

Poskytuje možnosti ako sú:

- **software development kit** – povolujúci integráciu FormDocs fillera s aplikáciou písanou v jazyku, ako napr. C++ alebo Java
- konverziu PDF súborov do FormDocs
- import alebo export z CSV, ASCII Text, ODBC, XML alebo zakódovanej url
- využívanie FormDocs v prostredí internetu
- pripojenie FormDocs filler na dátový zdroj

The screenshot displays the 'FormDocs Filler - Untitled3' application window. The menu bar includes 'File', 'Edit', 'View', 'Records', 'Tools', 'Window', and 'Help'. The toolbar contains icons for 'New', 'Open', 'Save', 'Send', 'Print', 'Spelling', and a zoom level of '91%'. A status bar at the bottom shows 'Page 1 of 1' and '<Natural Order>'. The main form area is titled 'WORK ORDER' and contains various input fields. On the left, there are fields for '[Your Company Name]', '[Company Address]', '[City, State, Zip]', and '[Phone Number]'. Below these are fields for 'Name:', 'Company:', 'Address1:', 'Address2:', 'City:', 'State/Prov.:', 'Country:', 'Zip/Postal Code:', 'Telephone:', and 'Fax No.:'. On the right, there are fields for 'Serial Number:', 'Customer Order Number:', 'Date:' (with the value '7/11/2002'), 'Job Site Telephone Number:', and 'Date of Order:'. At the bottom, there are fields for 'Job Number:', 'Job Name:', 'Job Location:', 'Date Started:', 'Terms:', and 'Order Taken By:'. A section titled 'PARTS AND MATERIALS' is visible at the very bottom of the form area.

Obrázok 1 - FormDocs

Formatta

Je spoločnosť poskytujúca webovo založený software na tvorbu elektronických formulárov. Ponúkajú systémy Design + Create na vytváranie a vyplňanie Complete + submit. Ponúka väčšinu funkcií ako ostatné existujúce systémy aj s možnosťou pripojenia sa na ODBC a JDBC povolujú bezpečne zdieľať formové dáta naprieč organizácie webových servisov a iných funkcií. [5]

Obrázok 2 - Formatta

goforms

Je jednoduchý nástroj určený prevažne na tablety a mobilné zariadenia za pomoci ktorého je možné využívať už existujúce formuláre pričom stačí načítať obrázok daného papierového formuláru do svojho profilu a jednoducho naň natáhať vstupné elementy a je ho možné vyplniť na každom mobilnom zariadení. [6]

Poskytuje možnosti ako sú:

- umožňujú jednoduché vytvorenie formuláru na tabletoch a mobilných zariadeniach
- zachytávanie podpisov na dotykových zariadeniach
- vykonávanie správ o vyplnených dátach užívateľmi



Obrázok 3 - goforms

1.3.2 Tuzemské společnosti

Najznámejším dodávateľom systémov pre elektronické dokumenty na našom území je spoločnosť Software602. Spoločnosť bola založená 1.1.1991 autori textového editoru Text602. Nástupom Microsoft office sa začala spoločnosť zameriavať na tvorbu interaktívnych elektronických dokumentov, ktoré sa stali základom siete Czech POINT. Od roku 2013 uvoľnila spoločnosť svoj produkt FormSigner pre všetky mobilné systémy. Najznámejšími produktami firmy sú FormDesigner a FormFiller. [7]

FormDesigner

- umožňuje návrh formulára od ruky či pripojenie dátového zdroja
- práca vo WYSIWYG režime
- vyžaduje znalosť XML dátových štruktúr a XSD schémy
- vytváranie webových formulárov
- používanie pokročilých funkcií a zložitých výpočtov
- knižnica komponent pre rýchly návrh

FormFiller

- určený na stolové počítače
- voľne šíriteľný
- možnosť využiť na fakturáciu inteligentných formulárov s podporou formátu ISDOC
- odosielanie formulárov emailom

100% Vyplněna_eFaktura.zfo - Software602 Form Filler

Domů

Schránka Úpravy Data

Doklad: Běžná faktura
Číslo dokladu: FVO S-7/2010

Dodavatel:
 Adresa sídla:
 Společnost a.s.
 Nováková 15, 140 00 Praha 4
 Česká republika
 Zápis v OR:
 Datum zápisu do OR:
 IČO: 63078236 DIČ: CZ63078236
 Kontakt:
 Telefon: 222 011 214
 E-Mail: info@602.cz
 Fakturační adresa

Odběratel:
 Adresa sídla:
 Company Inc.
 Na Žertvách 23, 180 00 Praha 8
 Česká republika
 IČO: 25625632 DIČ: CZ25625632
 Kontakt:
 Telefon:
 E-Mail:
 Fakturační adresa
 Dodací adresa

Ohrada: Bankovním převodem
 Banka: Česká spořitelna a.s. - běžný
 Číslo účtu: 0 0 0 0 0 0 - 0 0 0 0 9 7 6 5 5 2
 Kód banky: 0800
 IBAN: CZ04 0800 0000 0000 0097 6652
 SWIFT: GIBACZPX

Datum vystavení dokladu: 20.12.2010 Variabilní symbol: 11537
 Datum zdanitelného plnění: 25.1.2011 Konstantní symbol:
Datum splatnosti: 1.2.2011 Specifický symbol:

Zobrazit detaily dodacího listu

Zobrazit detaily objednávky

Poznámka k dokladu:
Vyhodnocení reklamní kampaně je přílohou faktury

☒ Plátce DPH ☒ Měna CZK
☐ Neplátce DPH ☐ Měna CIZÍ

| Kód | Název | Množství | Jednotková cena bez DPH | Sazba DPH (%) | Základ daně | Částka DPH | Celkem s DPH |
|----------|---|----------|-------------------------|---------------|-------------|------------|--------------|
| 20200001 | Fakturuje me Vám náklady na společnou reklamní kampaně v období | 1 M.J. | 2 000 | 20 | 2 000 | 400 | 2 400 |

Vyplněna_eFaktura.zfo [X]

1 / 2 PŘES Zkontrolovat formulář

Obrázok 4 - FormFiller

2 Špecifikácia zadania

Pred začatím skúmania akéhokoľvek riešenia je potrebné stanoviť jasné ciele práce. Výsledkom má byť systém, ktorý ma slúžiť na vytváranie inteligentných formulárov s jednoznačne definovanými vlastnosťami a nástrojom, za pomoci ktorého budú môcť užívatelia systému formulár využívať. Aby bolo možné dosiahnuť požadovaný výsledok, je potrebné stanoviť presný formár dokumentu, ako budú dáta ukladané. Následne vytvoriť systém schopný práce s týmto súborom.

2.1 Základný popis a ciele

Navrhovaný systém má spĺňať základnú podmienku – dokument, ktorý je výsledkom má mať vzhľad klasického papierového formulára pripraveného k tlači, musí s ním byť však asociovaná logika, ktorá plní úlohu sprievodcu pri vyplňaní formulára.

Primárnym riešením je vytvorenie modulu, pracujúceho nad dátmi užívateľa čím systém odbremení tvorca dokumentu poznať štruktúru formulára a poskytne mu možnosť zamerať sa výlučne na tvorbu vizuálnej stránky a príslušných výpočtov. Je však potrebné aby poznal aspoň štruktúru zdroja pri písaní výpočtov aspoň trochu, z dôvodu že systém podporuje vytváranie rovnako pomenovaných elementov na rôznych úrovniach.

Riešenie je jednoduché – je potrebné zabezpečiť, aby bola oddelená dátová časť dokumentu od vizuálnej, dátová časť bude tvorená užívateľom, alebo prevzatá z iného zdroja a previazaná s vizuálnou časťou. Týmto spôsobom zabezpečíme, že jeden dátový zdroj je použiteľný pre viacero dokumentov a jednoduché prevedenie úprav vo vizuálnej časti dokumentu bez zložitého zasahovania do dátovej časti. Výsledné riešenie by malo obsahovať modul pracujúci nad XML dokumentom ako v prehliadači tak aj na strane servera za pomoci XPath výrazov. Implementácia na strane užívateľa nám zabezpečí, že nie je potrebné blokovať užívateľské rozhranie pri každej zmene hodnoty rovnako aj zmeny závislých prvkov budú premietnuté okamžite. Server bude týmto odľahčený od neustáleho čítania súborov a kontroly dokumentu a jeho úloha sa obmedzí len na dobu ukladania a validáciu dokumentu.

Výpočty budú prebiehať pomocou XPath výrazov, ktoré umožňujú jednoduchú prácu s XML dokumentom. Tvorca tým pádom bude musieť poznať aspoň základy tvorby XPath výrazov, avšak aby nebol nútený k zložitým úkonom, bude výsledná aplikácia automaticky dopĺňať výrazy a pomáhať pri ich tvorbe. Tým pádom sa tvorba výrazov stane rýchla a priehľadná.

2.2 Kto bude s IS pracovať

Systém je tvorený ako viac užívateľský so zameraním na jednoduché používanie väčšiny užívateľov. Z tohto dôvodu sú vytvorené dve roly, v ktorých budú užívatelia v systéme vystupovať. Prvou rolou je rola User, ktorá reprezentuje bežného užívateľa systému prichádzajúceho vyplniť formulár. Druhou je administrátor, ktorý má hlavnú úlohu v správe systému. Bežný užívateľ môže ale aj nemusí mať práva na vytváranie nových dokumentov.

User

Tento typ užívateľa je každý platne zaregistrovaný užívateľ systému. Zameriame sa tu prevažne na ľudí v produktívnom veku, ktorý potrebujú uľahčiť prácu pri vyplňaní často rovnakých dokumentov. Systém ponúka užívateľom možnosť kontroly vstupných údajov, čo znamená že v prípade chyby nemusí formulár vyplňať nanovo, ako je to v prípade papierových tlačív a bude prebiehať automatický výpočet potrebných polí.

Podsystém vzdelávacích kurzov nám ponúka rovnako možnosť pod svojím účtom zúčastňovať sa kurzov, ku ktorým máme prístup. Tento typ užívateľa by mal mať možnosť vykonávať nasledujúce úlohy:

- prihlásenie a registrácia do systému
- vyhľadávanie formulárov
- otvárať prístupné formuláre
- zobrazit' vlastné už otvorené formuláre
- export formulára do podporovaného formátu
- zmazať vlastné dokumenty
- prihlásiť sa na kurzy
- odhlásiť sa z kurzov
- prístup k štúdiijným materiálom

Administrátor

Tento typ užívateľa bol do systému má v systéme vykonávať údržbu a takisto pridelovanie práv na tvorbu dokumentov.

Úlohy tohto užívateľa sú rozšírené o nasledujúce možnosti:

- otváranie nových a rušenie existujúcich kurzov
- pridávanie nových štúdiijných materiálov
- hodnotenie kurzov

Nie je žiadúce aby každý užívateľ, ktorý pristúpi do systému mal právo vytvárať nové formuláre. Systém by sa stal rýchlo zahlteným množstvom niekedy aj nekvalitných alebo aj duplicitných fomulárov a užívateľ by mal veľmi náročné vyhľadať želaný fomulár. Z toho dôvodu môže každý užívateľ systému otvárať existujúce dokumenty ale aby mohol formulár aj vytvoriť, musí mať nastavené práva *Creator*.

Definuje nasledujúce možnosti:

- vytvoriť nový formulár
- úprava už existujúcich formulárov
- validácia formulárov

2.3 Zoznam vstupov a výstupov systému

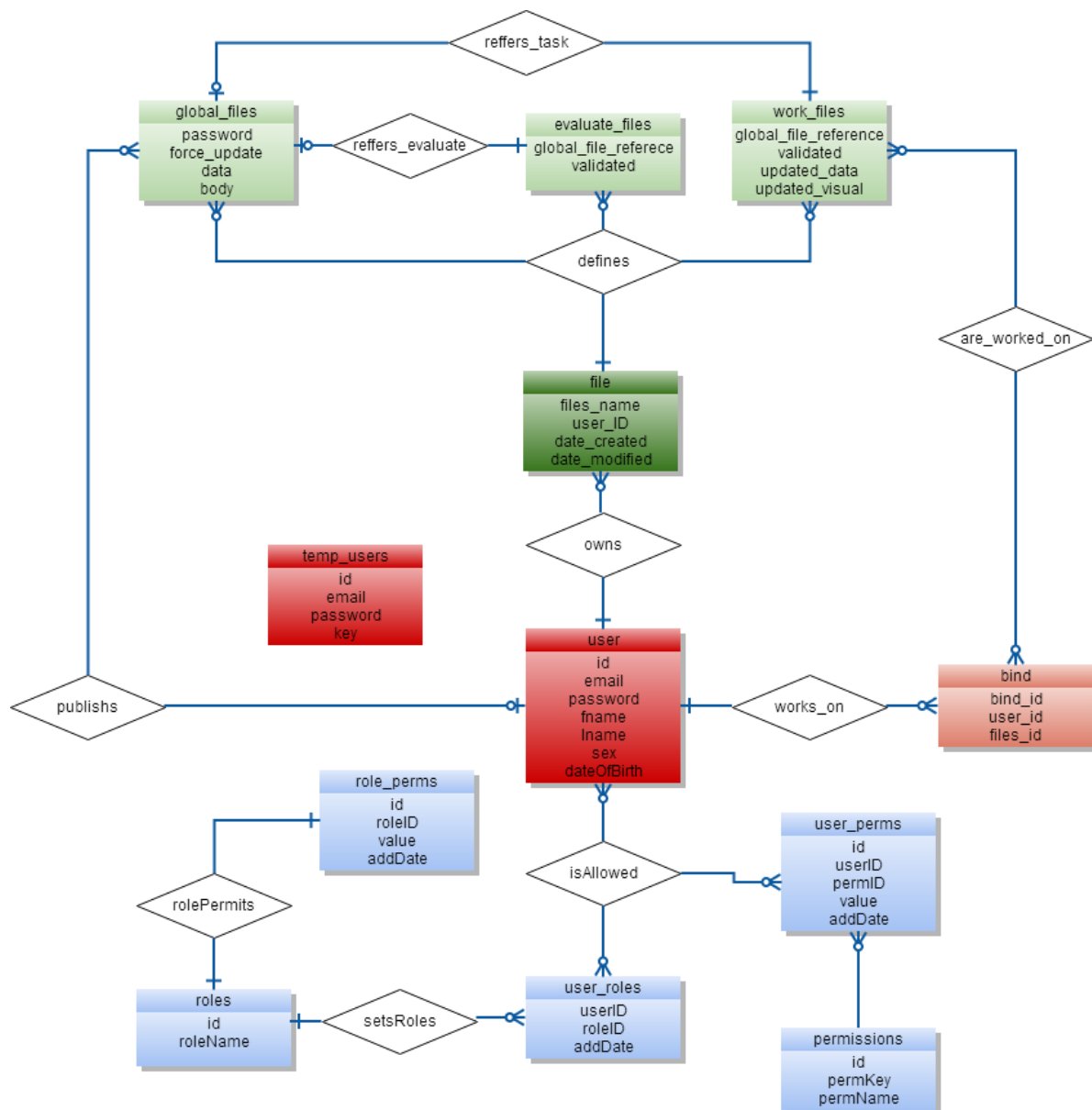
2.3.1 Vstupy systému

- Users – meno, prizvisko, email, heslo, pohlavie, dátum narodenia
- Pracovný súbor – menom dátátum vytvorenia, dátum poslednej zmeny, validný
- Dočasný užívateľ – email, heslo
- Globálne súbory – meno, dátum vytvorenia, dátum poslednej upravy, heslo, vynútenie aktualizácie
- Oprávnenia – meno
- Rola – meno
- Oprávnenia Roly – rola, Pridané
- Vyplňaný súbor – meno súboru, vytvorené dňa, modifikované, validný
- Užívateľské oprávnenia – Oprávnenie
- Užívateľské roly – rola, datum
- Subor – suboroveDáta

2.3.2 Výstupy

- Prehľad existujúcich dokumentov
- Zoznam rozrobemých dokumentov
- Prehľad otvorených kurzov
- Vyhľadávanie inteligentný formulár
- Otváranie Pdf súborov

2.4 ERD dieagram



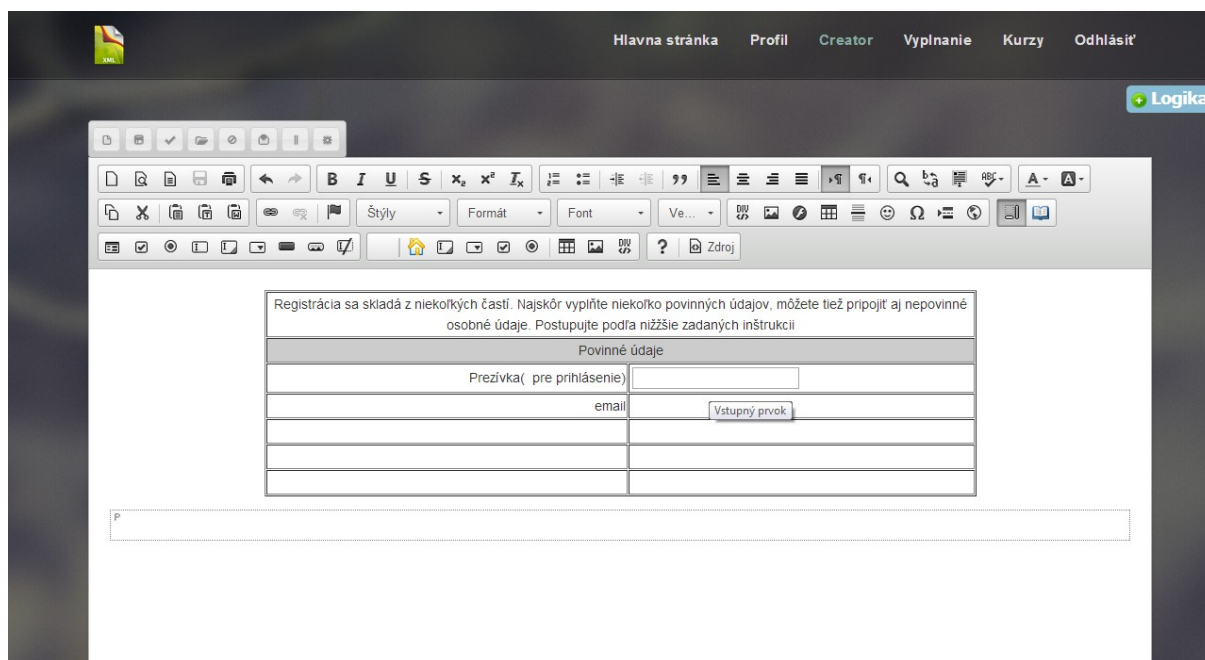
Obrázok 5 - ERD primárnej časti systému

2.5 Návrh užívateľského rozhrania

Primárnou úlohou systému je už tvorba alebo práca s formulármi a tým pádom užívateľa systému budú tráviť väčšinu času práve na stránke otvárajúcej formulár alebo dizajnéra formulára. Zvyšné časti majú skôr informačný, resp. doplnujúci charakter.

Tvorca formulárov

Stránka je zložená z dvoch základných častí, nutných pre prácu s dokumentom a prvou dominantou stránky je textový editor (CKEditor). Ten má za úlohu tvorbu vizuálnej stránky formuláru a následne ho previazať s dátovou pomocou prídavných rozšírení. Druhou, menej viditeľnou, časťou stránky je rozklikávací editor logiky, ktorý má za úlohu prácu s dátovou a logickou vrstvou dokumentu vo forme stromovej štruktúry.



Obrázok 6 - Tvorca formulárov

Čítač dokumentov

Najpodstatnejšia, ale zároveň z užívateľského hľadiska najjednoduchšia časť systému. Jednoduchá je z toho dôvodu, pretože práca užívateľa spočíva mimo klasických súborových operácií, len vo vyplňovaní formulára. Jednotlivé vstupné polia sú automaticky overované a výpočty sú prevádzane okamžite.

Hlavná stránka

Úvodná stránka plní informačný charakter s možnosťou vyhľadávania formulárov a ich automatického otvorenia. Je tu aj panel zobrazujúci päť naposledy otvorených spolu s piatimi poslednými vytvorenými formulármi.



Obrázok 7 - Hlavná stránka

Profil

Na stránke profilu sú zobrazené základné informácie užívateľa s možnosťou úpravy a takisto je tu umiestnený aj zoznam užívateľových dokumentov.

Prihlásenie

Prihlasovaciu stránku tvorí klasický prihlasovací formulár, kde sa užívateľ môže prihlásiť pomocou emailu a hesla.

Registrácia

Registrácia je vstupným dobom do systému. Po registrácii je užívateľovi zaslaný overovací email a po verifikácii emailu je mu priradená rola. Na registráciu je potrebný len platný email a heslo.

Edukačný podsystém

Edukačný podsystém obsahuje množstvo statických stránok slúžiacich na množstvo úkonov. Patria tu stránky ako správa kurzov, vytváranie nových kvízov ale ponúkajú aj využívanie nami definovaného súboru za účelom výučby.

2.6 Použité technológie

Pri postupe a návrhu nášho riešenia, sme zohľadňovali základný návrh a potreby systému, na základe čoho sme zvolili použitie rôznych technológií, popísaných v nasledujúcich podkapitolách.

2.6.1 Serverové jadro

Na strane servera sme používame jazyk PHP a framework CodeIgniter, zabezpečujúci autentifikáciu údajov, registráciu, export formulára, čítanie dát z databázy MySQL alebo ukladanie na súborový systém. Dôvodom zvolenia frameworku CodeIgniter je fakt, že nieje typický framework, kvôli jeho veľkosti. Vďaka tomu, že je tak malý zohráva rýchlosť interpretácie kódu významnú rolu a zanecháva nízku stopu. CodeIgniter je založený na návrhovom vzore MVC.

2.6.2 Použitá databáza

Na uchovávanie dát sme zvolili databázový server MySQL. Ukladáme v nej užívateľov, súbory, dočasných užívateľov čakajúcich na potvrdenie a iné. Dôvodoch na zvolenie MySQL databázového servera je hneď niekoľko. Podporuje uložené procedúry, transakcie a nepredpokladá sa veľký objem dát. Zároveň je MySQL najpoužívanejšou databázou u frameworku CodeIgniter.

2.6.3 Ajax, JQuery a javascript

Použitie JQuery frameworku na vykonávanie bežných operácií na stránke zameraných zväčša na spúšťanie udalostí, či použitia Ajax na dotazovanie sa na server je samozrejmosťou ako takmer v každej webvej aplikácii. Je však dôležité, že primárnu úlohu v celom systéme na seba preberá jazyk javascript, implementujúci modul vykonávajúci všetky dôležité operácie ako úprava XML súboru či spúšťanie XPath výrazov.

3 Formát a modul AIDoX

Rovnako, ako množstvo iných kancelárskych nástrojov, je aj naše riešenie založené na formáte XML. Súčasťou spôsobu ako budeme s inteligentnými formulármi pracovať je stanovenie výsledného formátu dokumentu. Ak by sme tak neučinili bol by náš výsledný dokument len obyčajný XML súbor držiaci nejaké dáta. Stále to však nestačí, XML nevykonáva žiadne operácie nad dátami a preto musíme zabezpečiť aby boli pravidlá nami definovaným formátom prevedené aj na funkčnú úroveň pre jednoduchú prácu ako na strane užívateľa tak aj servera. Nasledujúca kapitola predstavuje nami implementované riešenie, spolu so stručným popisom základných pojmov.

3.1 XML

XML (Extensible Markup Language) je značkovací jazyk odvodený od staršieho formátu SGML (ISO 8879). Na rozdiel od SGML je mnoho parametrov predom určených a nejde im meniť maximálnu dĺžku názvov značiek, použité oddeľovače a špeciálne znaky atď. XML už rovno počíta s podporou všetkých možných jazykov, takže není tak úzko zviazaný s angličtinou ako väčšina predchodcových počítačových technológií. Navyše je syntaxa zápisu dokumentu v XML oproti SGML pomerne prísna, čo umožní omnoho ľahší a lacnejší vývoj aplikácií, ktoré umožňujú s týmto jazykom pracovať.

Ako už bolo spomenuté prísna syntaxa jazyka, jednoduchosť a silná podpora cez Unicode, ho robí vysoko flexibilným a použiteľným v mnohých sférach informačných technológií. Primárnym aspektom pre našu prácu je fakt, že XML formát sa stal základným formátom pre kancelárske nástroje ako sú Microsoft office, či Openoffice.org.

XML dokument pozostáva z troch primárnych častí, ktorými sú:

- a) Deklarácia – je procesová inštrukcia identifikujúca dokument ako XML. Definuje 3 atribúty:
 - version – špecifikuje verziu XML štandardu
 - encoding – špecifikuje kódovanie dokumentu (napr. UTF-8)
 - standalone – ak je nastavené áno, tak dokument má interné DTD a ak nie, je nalinkovaný na externé DTD

`<?xml version="1.0" encoding="UTF-8" standalone="no" ?>`

Obrázok 8 - XML definícia

- b) Element – každý XML dokument je zložený z jedného a viac, do seba vhlbených elementov, ktoré sú vyznačené pomocou tzv. tagoch. Názvy jednotlivých tagoch sú ohraničené

začiatočným (<) a koncovým (>) znakom. Obsah jednotlivých elementov je vložený medzi tagy elementu, pričom ukončovaci tag je doplnený o ukončovaci znak (/). Prázdné elementy môžu pozostávať len zo skráteného zápisu využitím len ukončovacieho tagu. XML striktne vyžaduje, aby ku každému začiatočnému tagu existoval korešpondujúci ukončovaci.

- c) Atribút – Každému elementu v XML môže byť pridružené ľubovoľné množstvo atribútov, ktoré majú za úlohu zväčša pripojenie rôznych metainformácií k elementu. Atribút je vložený do tagu a pozostáva z páru názov/hodnota, kde každý atribút môže držať práve jednu hodnotu. Ak je potrebné aby držal viac hodnôt, je potrebné stanoviť akýsi oddeľovač, ktorým býva zväčša bokočiarka alebo prázdny znak.

<element id="2" name="attr">Element hodnota</element>

Obrázok 9 - XML element a atribúty

3.2 Jazyk XPath

V predchádzajúcej podkapitole sme sa dozvedeli, že dáta budú ukladané vo formáte XML a spôsob akým budeme prístupovať k informáciám uloženým v ňom je pomocou jazyka XPath.

„XPath je W3C štandard jazyka používaný na vyhľadávanie informácií v XML dokumente.“. Je fundamentálnou časťou technológií ako sú XSLT alebo aj subjektoch ako XQuery alebo XPointer. Jeho úloha spočíva v definovaní cesty k elementom. Nie je však pravidlo, že výsledkom výrazu je vždy práve jeden jedinečný uzol. V XML dokumente často zodpovedá jednému výrazu viacero elementoch a môže vrátiť kolekciu uzlov. Pomocou jazyka XPath je však možné docieľiť aj vykonávanie logických či matematických výrazoch nad dokumentom. Výsledkom takéhoto výrazu je napríklad číslo alebo logický výraz. Toto je dôležité pre našu aplikáciu, keďže väčšina výpočtov je riešená práve týmto spôsobom a tvorca formulára bude definovať výpočty práve v tomto jazyku.

Syntax XPath výrazu je jednoduchá. Skladá sa z takzvaných "lokačných krokov", čo sú v skutočnosti názvy elementov predelené lomítkami. Kontextový uzol zas určuje miesto v dokumente odkiaľ bude výraz vykonávaný. V našej aplikácii práve tento fakt zohráva dôležitú úlohu pri používaní opakovateľných sekcií. Nebolo by efektívne aby užívateľ zadával miesto z akého sa bude výraz vykonávať. Systém je schopný sám rozpoznať element od akého začať výpočet.

3.3 Formát AIDoX

Aby bolo možné pracovať je potrebné najskôr definovať ako formát AIDoX plní túto úlohu a zabezpečuje konzistentnosť údajov. Pri ukladaní dát musí byť oddelená dátová časť od vizuálnej. Chceme aby bolo možné na rovnakú dátovú vrstvu naviazať viacero dokumentov, alebo upravovať vizuálnu stránku bez nutnosti vstúpiť do dátovej časti. Toto rozdelenie nám zabezpečí možnosť, vytvoriť jednu dátovú štruktúru pre sériu dokumentov, v ktorej by bola následne asociovaná len jedna vizuálna stránka poprípadе nadviazaná na ňu.

Dáta sú ukladané vo formáte XML rozdelené do troch základných blokov. Jednotlivé názvy zodpovedajú úlohe akú zohrávajú.

3.3.1 Model

Model je dátová sekcia, ktorá má za úlohu ukladať informácie, ktoré sú vstupnými dátmi užívateľa. Samotná model sekcia je obsiahnutá v controller sekcii. Dôvodom je fakt, že controller je závislá od štruktúry modelu a musí ju reflektovať. Táto sekcia nekladie obmedzenie na množstvo užívateľom definovaných elementov, avšak nie je povolené vytvárať na rovnakej úrovni dva rovnaké elementy s rovnakým menom, k tomuto účelu slúžia opakovateľné sekcie z dôvodu jednoznačnej rozlíšiteľnosti jednotlivých elementov.

Zoznam povolených atribútov

| Názov | Hodnota | Popis |
|-------|-------------------|---|
| id | numerická hodnota | id atribút je špecifikovaný u opakovateľných elementov na jednoznačnú identifikáciu |

Tabuľka 1 - model atribúty

3.3.2 Controller

Táto sekcia obsahuje len elementy s názvom bind. Controller tvorí a určuje ako bude s dátmi v model sekcii narábané. Štruktúra sekcie je identická so štruktúrou model sekcie vďaka čomu asocjuje logiku na jednotlivé elementy a pomocou *element_Xpath* atribútu odkazujúceho na konkrétny element. Každý bind musí obsahovať jedinečný id atribút, ktorý sa vo väčšine prípadov rovná názvu prislúchajúceho elementu na ktorý ukazuje. Toto však nie je pravidlo, pretože formár povoľuje rovnaké pomenovania elementov na rôznych úrovniach. Z toho dôvodu je zavedený name atribút obsahujúci názov elementu, na ktorý je odkazované. Špecifickým prípadom sú opakovateľné elementy, ktoré zdieľajú všetky jeden bind element, cez ktorý k dátam pristupujú pomocou platného identifikátoru.

Niektoré povolené atribúty:

| Názov | Hodnota | Popis |
|----------------|-----------------------------------|---|
| id | akýkoľvek platný názov elementu | vo väčšine prípadov je zhodný s názvom elementu na ktorý odkazuje, v prípade existencie viacerých elementov je náhodne generovaný |
| name | názov elementu, ktorý sa odkazuje | vždy zhodný s názvom odkazovaného elementu |
| element_XPath | platný XPath výraz | obsahuje XPath výraz odkazovaného elementu |
| doNotCalculate | platný XPath výraz | ak vráti false alebo žiaden výsledok |
| calculate | platný XPath výraz | vráti novú hodnotu elementu |
| repetition | platný počet oddelený čiarkami | nastavuje minimálny, maximálny a štartovací počet elementov (napr. min:1, startUp:3,max:5) |

Tabuľka 2 - niektoré bind atribúty

3.3.3 View

View sekcia je vizuálna stránka dokumentu obsahujúca validný XHTML dokument. Prislúchajúce vstupné prvky dokumentu sú previazané s dátovou pomocou id bind elementov. Bind tým tvorí most medzi oboma sekciami ako bude s dátami manipulované. Rovnako ako dátova sekcia tvorená controllerom a modelom je aj táto sekcia nezávislá. To znamená, že môže byť použitá pre viacero zdrojov. Toto je možné vďaka tomu, že view tvorí v skutočnosti šablónu na základe ktorej je dokument zostavený.

3.4 AIDoX modul

Ako sme uviedli v predchádzajúcej podkapitole prístup k dátam bude realizovaný pomocou XPath výrazov. Formát AIDoX je založený na formáte XML, ktorá vznikla za účelom štruktúrovania, ukladania a transportu dát. Formát sám teda nevykonáva žiadne operácie. Na realizáciu týchto operácií sme na strane užívateľa vytvorili javascriptový modul pod názvom AIDoX. Úlohou modulu je realizácia požadovaných operácií na otvorenom formulári vo webovom prehliadači. Jeho súčasťou sú dva objekty slúžiace na prácu s XML súborom a dva prídavné moduly, pracujúce s dokumentom v stanovenom formáte.

Samotný AIDoX modul využíva primárne dve vstavané funkcie ktorými sú *OpenNew* slúžiaca na otvorenie nového dokumentu, uložením jeho instance do premennej *instances* a funčia *CloseDocument* na zatvorenie dokumentu. Samotná premenná *instances* je pole ktoré umožňuje modulu otvárať viacero dokumentov súčasne.

3.4.1 AIDoX.wrapper

Problémom s výsledkom navráteným po navrátení XPath výrazu je ten, že manipulácia s ním je pomerne zložitá kvôli množstvu variácií. Na elimináciu tohto problému slúži objekt AIDoX.wrapper. Jej názov napomína čo je jej úlohou. Nová instance tohto objektu, je navrátená po vykonaní funkcie *Query*, ktorá je popísaná v nasledujúcej podkapitole. Výsledok tejto funkcie je predaný konštruktoru objektu.

Novovytvorená instance nám teraz ponúka možnosť iterácie cez výsledok pomocou funkcií ako sú *next*, *current*, *count* či *first*. Takáto funkcionality by však bola redundantná, pretože podobné možnosti nám ponúka aj objekt navrátený vstavanými funkciami. Význam wrappera spočíva v schopnosti analyzovať výsledok a navrátiť požadovanú hodnotu alebo uzol. Ak by sme pracovali v prostredí webového prehliadača, ktorý využíva na vykonávanie XPath výrazov funkciu *document.evaluate*, tak výsledok vykonaného výrazu je objekt XPath result, ktorý však ešte rozlišuje typy výsledkov „resultType“.

Príklad možných typov výsledkov:

| Typ výsledku | Hodnota | Popis |
|------------------------------|---------|---|
| ANY_TYPE | 0 | Vráti akýkoľvek výsledok aký vyhovuje. |
| NUMBER_TYPE | 1 | Výsledok výrazu obsahuje jediné číslo. |
| UNORDERED_NODE_SNAPSHOT_TYPE | 6 | Vráti sadu uzlov so snímku všetkých vyhovujúcich výrazov. |
| FIRST_ORDERED_NODE_TYPE | 9 | Vrátený je prvý uzol vyhovujúci výrazu. |

Do kódu by sme tým pádom museli vždy v mieste volania rozlišovať požadovaný typ explicitne a v prípade zmeny typu výsledku upraviť kód na každom mieste volania. Wrapper tento výsledok analyzuje a vráti nám požadovanú hodnotu alebo uzol bez toho aby sme v mieste volania museli niečo špecifikovať. Vďaka tomuto môžeme s výsledkom pracovať a dostaneme späť vždy hodnotu v očakávanom type. Funkcia *current* a *first* slúžia v prípade navrátenia hodnoty alebo len jedného uzlu úplne identickému účelu.

3.4.2 AIDoX.xml

Predchádzajúca podkapitola nám priblížila AIDoX.wrapper objekt, za pomoci ktorého sme schopný pohodlne pracovať s výsledkom XPath výrazu. Aby sme však instanciu tohto objektu mohli vytvoriť, je potrebné definovať objekt, ktorý bude nad XML súborom pracovať. Na túto úlohu existuje objekt AIDoX.xml, ktorý v sebe združuje bežné operácie pred XML dokumentom ale aj vykonáva požadované XPath výrazy, konštruktor prijíma XML dokument buď ako objekt alebo textový reťazec, ktorý je následne parsovaný. Prídavné moduly popísané v nasledujúcich podkapitolách ho využívajú na úpravu štruktúry alebo vyplňanie. Obsiahnuté funkcie ako insert, update ci delete spolu s ich variáciami plnia bežnú úlohu pri tvorbe dokumentu. Problémom je však vyhľadávanie, keďže nie všetky webové prehliadače ponúkajú jednotný spôsob vykonávania XPath výrazov. Na tento účel slúži funkcia Query a vracia výsledok na základe webového prehliadača v akom práve pracuje. Pri vykonávaní funkcií ako *Get* alebo *GetID* je volaná následne vždy táto funkcia a výsledkom je už spomínaná nová instancia objektu AIDoX.wrapper.

3.4.3 AIDoX.format

Objekty definované v predchádzajúcej časti definovali spôsob práce s dokumentom, tie nám však dovoľia pracovať s akýmkoľvek XML dokumentom. Tento prídavný modul plní dôležitú úlohu pri tvorbe dátovej časti formulára. Funkcie majú za úlohu vytvárať stanovený formát na základe definovaných pravidiel. Obsiahnutá funkcia *AddElement* vytvorí element v model časti dokumentu a k nemu odpovedajúci bind element s atribútom id, name a element_XPath s korešpondujúcimi hodnotami. Musí však zabezpečiť, aby nebolo možné vytvoriť dva elementy s rovnakým názvom na tej istej úrovni, z dôvodu kolízie názvov, na tento účel sú vytvorené opakovateľné sekcie. Funkcia však povoľuje vytváranie elementov s rovnakým názvom na rovnakých úrovniach. Funkcia síce túto možnosť ponúka, žiadnym spôsobom to negatívne neovplyvní funkcionality, funkcia však každému elementu vytvára korešpondujúci bind element, ktorého id atribút je vo väčšine prípadov meno elementu na ktorý odkazuje. V prípade existencie prvku s takýmto názvom je id vygenerované čo spôsobuje nepriehľadnosť. Z toho dôvodu je lepším postupom pridávanie takýmto elementom prefix názvu alebo skratky názvu predka.

Pri volaní funkcie *RemoveElement* je dôležitý fakt, že nie je povolené odstánenie koreňového elementu, úprava názvu však áno. Funkcia *RenameElement* upraví názov elementu spolu s element_XPath atribútom, ako aj všetkých odvodených. Táto funkcia má jednu veľkú nevýhodu. Zmena názvu sa premietne na všetky odvodené elementy, ale zmenou cesty k elementom sa znefunkčnia všetky už existujúce výpočty založené na starých XPath výrazoch. Z toho dôvodu by mala byť štruktúra dátovej časti už napred známa. Pri nastavovaní atribútov bindovacieho elementu je volaná funkcia *UpdateElementLogic*, ktorej úlohou je nastavenie nového správania tohto prvku na stanovené hodnoty. Sem sa ukladajú zväčša XPath výrazy definujúce správanie formulára.

3.4.4 AIDoX.evaluate

Tento prídavný modul je poslednou časťou AIDoX modulu. Prídavný modul opísaný v predchádzajúcej podkapitole má za účel tvorbu platnej dátovej časti formulára. Úlohou tohto modulu je práca s touto dátovou časťou. Automaticky pri otvorení nového dokumentu sa volajú funkcie *NewEvaluateDoc* a *PrepareDocument*. Sila funkcie *NewEvaluateDoc* spočíva v tom, že vyhľadá opakovateľné sekcie v dokumente a namapuje ich do poľa *repetitionSections* pod ich príslušným id. Tento krok zabezpečí, že pri tvorbe nových opakovateľných sekcií nie je potrebné zakaždým generovať ich štruktúru nanovo a stačí vytvoriť len ich kópiu a nastaviť ich príslušným identifikátorom. Inak by bolo potrebné opakovane rekurzívne prechádzať dokumentom s rovnakým výsledkom len množstvom redundantným krokov. Funkcia *PrepareDocument* následne dokument zostaví. Po zostavení dokumentu môže užívateľ začať pracovať, ak sa rozhodne pridať novú sekciu v dokumente, automaticky skript *AIDoX_visual.js*, popísaný v nasledujúcej podkapitole, začne zisťovať či je možné vytvoriť novú sekciu volaním funkcie *AddSection*. Táto funkcia na základe príslušného bind elementu zistí, či je možné vytvoriť ďalší, ak áno, funkcia vytvorí nový element zo zálohy vytvorenej pri otváraní dokumentu. Následne sa vygeneruje nové id a nastaví sa. Funkcia prijíma takisto index elementu, za ktorý má byť vložený nový prvok. Ak nie je zadaný alebo neexistuje je pridaný na koniec. Pri odstraňovaní sekcie sa klasicky maže element, musí však platiť, že nový počet prvkov nebude menší ako zadané minimum.

Pravdepodobne najdôležitejšou funkciou je funkcia *Update*. Pri zmene akéhokoľvek vstupného elementu vo formulári je volaná práve táto funkcia. Funkcia začne postupne overovať správnosť všetkých potrebných výpočtov uložených v atribútoch príslušného bind elementu ak existujú. To znamená, že ak by bola hodnota atribútu nastavená na `false()` zmena hodnoty by nikdy nemohla prejsť. Ak by však bola hodnota XPath výrazu napríklad `/faktura/celkovenaklady > 0` a výsledok by bol `true`, podmienka je splnená. Ak však hodnota prejde neznamená to, že je automaticky platná na elemente môžu byť závislé iné elementy a z toho dôvodu je volaná funkcia *CheckDependencies* a predá jej zoznam závislých elementov. Tá začne postupne volať na každý jeden funkcia *Calculate*, ktorá vypočíta novú hodnotu a overí jej platnosť. Ak je hodnota platná, overí sa ďalší prvok, ak nie automaticky oznámi funkcia *CheckDependencies* zlyhanie *Update* funkcii a ten sa postará o obnovu dát do pôvodného stavu. Ak sú však platné všetky hodnoty *Update* vráti pole obsahujúce nové hodnoty.

Funkcia *Calculate* logicky takisto volá funkciu *CheckDependencies* pre svoje závislé prvky. Aby tu nevznikol problém kruhovej závislosti sú takéto formulárové dokumenty označené ako nie validné už pri zostavení.

3.5 AIDoX_visual.js

Tento jednoduchý skript je založený na JQuery frameworku a slúži na zabezpečenie vizuálnej komunikáciu užívateľa s formulárom, keďže modul AIDoX sa stará o vnútornú integritu dát. Žiadnym spôsobom však neovplyvňuje vizuálne zobrazenie. Nadviazaním na príklad volania funkcie *AddSection* z predchádzajúcej podkapitoly je táto funkcia volaná vždy ako užívateľ klikne na tlačidlo pridania novej sekcie. O spustenie tejto udalosti sa stará práve tento skript, musí však dbať na to, že ak sa vytvorí sekcia obsahujúca ďalšie opakovateľné elementy musia byť nastavené ich identifikátori hodnotami navráteními funkciou *AddSection*.

Aby nebol formulár preplnený zbytočnými tlačidlami zobrazí skript tlačidlá na pridávanie a odoberanie sekcie len keď je myš umiestnená nad ňou. Toto je však problém, ak je opakovateľná sekcia prázdna. Skript sa z toho dôvodu stará o to aby pripol každej takto prázdnej opakovateľnej sekcii tlačidlá do doby kým nebude existovať aspoň jedna sekcia. Základnou úlohou je validácia vstupných údajov užívateľa. Na tento účel využíva funkciu *track* pluginu *verify.js*. Funkcia jednoducho pri každej zmene hodnoty overí platnosť volaním spomínanej funkcie *Update* a rozhodne či vykonať požadované zmeny alebo obnoviť pôvodné hodnoty.

4 Implementácia

4.1 Použité externé nástroje

Značná časť bakalárskej práce je zostavená aj nie nami impementovanými nástrojmi. Tie sa starajú v prevažnej väčšine o vizuálne zobrazenie dát a sú to rôzne malé JQuery pluginy. Patrí tu však aj napríklad textový editor CKEditor, zohrávajúci výraznú úlohu pri vizuálnom návrhu formulárov.

CKEditor

Pre naše riešenie je dôležité zahrnutie HTML textového editora za pomoci ktorého je možné zostaviť vzhľad dokumentu vo WYSIWYG režime. Práve tomuto účelu slúži CKEditor, ktorý bol dizajnovaný z dôvodu sprístupnenia výhod textového procesora pre prácu vo webovom prehliadači. Jeho jednoduchosť, množstvo rozšírení a kompatibilita s väčšinou prehliadačov ho robí jedným z najpoužívanějších textových editorov v prostredí internetu.

Z nášho hľadiska ho robí dôležitým jeho jednoduchá rozširiteľnosť o pluginy. Bolo vytvorených niekoľko pluginov na správu elementov vstupných dát, ako input pre užívateľa, aby bolo možné dokument naviazať na dátový zdroj. Tieto pluginy rozširujú CKEditor o volania funkcií AIDoX modulu.

Verify.js

Tento jednoduchý ale zároveň mocný a flexibilný plugin sa používa na vyhodnotenie vstupných údajov užívateľa automaticky po vykonaní zmeny. Pôvodne sme stavali aplikáciu na JQuery Validation Engine, tá však nebola veľmi vhodná pre naše účely pretože jej implementácia bola veľmi zložitá pre množstvo chýb. Neskôr bol však nahradený práve týmto pluginom ktorý ponúka tvorbu vlastných pravidiel k už existujúcim. V našej implementácii využívame najvýraznejšie funkciu *track*, ktorá je vyvolaná pri zmene akéhokoľvek vstupného prvku.

Notify.js

Je to JQuery plugin slúžiaci jednoduchému zobrazovaniu upozornení užívateľovi za pomoci notifikácií.

jsTree

Tento JQuery plugin zohráva dôležitú úlohu pri vykresľovaní dátovej sekcie za pomoci interaktívnej stromovej štruktúry. Ponúka mnoho výhod ako rozširiteľnosť, nastaviteľnosť či čítanie dát pomocou Ajax. Pre účel našej práce je významná hlavne pre možnosť naviazovania funkcií spätných volaní na kontextové menu tejto stromovej štruktúry. Toto nám ponúka prehľadnú a rýchlu prácu.

HighCharts

Aplikácia využíva na prácu s grafmi na JavaScriptoch highcharts knižnicu, založenú čisto na javascriptoch, vďaka jeho jednoduchosti a flexibilitě.

dompdf

Je HTML na PDF konvertor napísaný v programovacom jazyku PHP, spravujúci prevažne CSS 2.1 a niektoré vlastnosti CSS3. Je tu pomerne silná limitácia pri práci s nie validným HTML dokumentom. CKEditor však tento problém rieši za nás a ak by aj užívateľ poskytol nevalidný dokument systém ho odmietne uložiť.

4.2 Aplikácia na strane užívateľa

V predchádzajúcej kapitole sme vytvorili modul za pomoci ktorého sme schopný spravovať nami otvorený dokument, a v predchádzajúcej podkapitole sme si priblížili prídavné nástroje použité v našom systéme. Funkcionalitu týchto prvkoch následne využijeme na oboch nástrojoch praujúcich s dokumentom. Tvorca formulárov aj Čítač formulárov využívajú oba nami definovaný modul AIDoX. Oba však vykonávajú rozdielne operácie.

Tvorca formulárov

Tvorca formulárov vytvára výsledný formulár na dvoch úrovniach. Aby bola možná práca pri zostavení stránky sa načítajú oba potrebné moduly AIDoX a CKEditor. Systém využíva súbory cookies na obnovenie posledného otvoreného súboru ak nebol zavretý. Po otvorení nového, resp. obnovení existujúceho, súboru pripraví AIDoX dátovú časť a CKEditor je za pomoci vstavanej funkcie *setData()* nastavený na vizuálnu časť formulára. Užívateľ môže teraz upravovať jednotlivé bloky podľa potreby.

AIDoX poskytuje aj pluginy pre CKEditor ktoré, zabezpečujú vkladanie elementov pre vstupné dáta, ako aj iných podporovaných ako sú napríklad opakované sekcie. Tieto pluginy si vyžadujú pri vytvorení každého nového prvku, nastavenie konkrétneho elementu do ktorého budú vstupnú hodnotu ukladať. Pri validácií budú obe časti spojené do jedného dokumentu na servery a ten overí, či sú všetky nastavené všetky požadované identifikatory. Túto nutnosť zadávať identifikátor je však možné vypnúť. Hodí sa to v prípade tvorby šablóny, pri validácií by však vstupné hodnoty bez identifikatora nepovolili jeho validáciu.

Jednotlivé výpočty zapisuje tvorca formulára do textového poľa umiestneného v dialógovom okne, po vybratí možnosti *Logika* v kontextovom menu daného elementu v stromovej štruktúre. Aby

užívateľ nemusel vyplňať celé výrazy ručne, povoľuje systém automatické dopĺňanie výrazov pre uľahčenie práce. Žiaľ javascript nepovoľuje XPath 2.0 a z toho dôvodu systém ponúka možnosť zostavenia niektorých funkcií ako je *max()* či *min()* v podporovanej verzii 1.0.

Po dokončení práce na formuláry, ho môžeme zvalidovať a ak ho server označí ako validný je, možné publikovať či už ako verejný alebo zaheslovaný formulár. K takýto publikovaný formulár môžu následne ostatní užívatelia vyplňať podľa potreby ak majú k nemu prístup.

Čítač formulárov

Čítač formulárov, rovnako ako tvorca, potrebuje na svoju činnosť modul AIDoX. Na správu vizuálnej časti ale využíva miesto CKEditora, skript *AIDoX_visual.js*, ktorý bol popísaný v predchádzajúcej kapitole.

Systém pri otvorení každého dokumentu vytvorí nový frame v ktorom otvorí dokument, zostaví ho a povolí vyplňanie. Tento frame je nastavený vždy na aktuálnu veľkosť dokumentu aby nevznikalo zbytočné preskakovanie obrazu. Otvorený formulár je možné vyplňať ako akýkoľvek papierový formulár, avšak na jeho vyplňanie nepoužívame pero ale klávesnicu. Po vyplnení dokumentu môžeme exportovať tento dokument ako PDF súbor alebo stiahnuť grafy v dokumentu keďže hightcharts túto možnosť povoľujú. Toto je prakticky všetko čo čítač vykonáva keďže jeho účel spočíva v otváraní a vyplňaní dokumentu.

4.3 Aplikácia na strane servera

V podkapitole použité technológie sme definovali, použitie jazyka PHP a frameworku CodeIgniter na strane servera. CodeIgniter je postavený na návrhovom vzore Model-View-Controller. Toto nám pomôže vybudovanie aplikácie oveľa rýchlejšie než ako by sme pracovali na nej od základov. Za najpodstatnejší aspekt využitia tohto frameworku sme považovali je veľkosť a tým súvisiacu zanechanú stopu. Náš systém má pracovať prevažne na strane užívateľa a tým pádom nie je potrebná veľká škála nadbytočnej funkcionality.

4.3.1 Model

Jednotlivé modely rerezentujú triedy ktoré pracujú s výsledkom v databázy. Vo frameworku CodeIgniter tieto triedy rozširujú triedu *CI_Model*. Nachádzajú sa v zložke *application/models/* a meno rozširujúcej triedy musí mať prvé písmeno veľké a zvyšok je malým. Modely nie sú povinné a využívame ich pri práci s databázou. V našich modeloch využívame Active Record so vstavanými funkciami pre prácu s databázou.

4.3.2 Controller

Controllery sú miesto kde sa rozhoduje ako bude naráňané s HTTP dotazmi na server. Meno controllera bude následne tým pádom asociované s URI. Všetky controllery sú obsiahnuté v priečinku *application/controllers/* a každý controller vo frameworku CodeIgniter rozširuje metódu *CI_Controller*. To znamená teda, že každý eden dotaz bude posilaný na konkrétny controller a ten môže využívať instance modelov na získavanie dát z databázy.

4.3.3 View

View sú zas proste len webové stránky, ktoré sú do controllera načítané a môžu aj samá načítavať iné view, ktoré majú za úlohu vrátiť a zostaviť stránku alebo len jej časť. Vytvárajú sa do priečinka */application/view/*. Môžu im byť predané aj parametre pretože sú to neni triedy ako v prípade modelu alebo controllera a sú vlastne PHP skripty obsahujúce HTML kód. Obsah týchto súborov bude to čo uvidí užívateľ na svojom prehliadači.

4.3.4 Library

Predchádzajúce tri podkapitoly opisujú hlavné stavebné bloky návrhového vzoru MVC. Náš systém je vytvorený na základe tohto návrhového vzoru vďaka prostrediu v akom pracuje. Pre nás je však zaujímavejší priečinok *application/libraries/*. Nachádzajú sa tu triedy ktoré majú za účel vykonávanie iných funkcií než správou nad databázou.

Cieľom bakalárskej práce však je vytvorenie nástroja na tvorbu štrukturovaných dokumentov, a preto, vytvorenie nástroja len na strane užívateľa nie je dostačujúce pretože náš systém tieto dáta od užívateľa prijíma a následne ukladá do príslušného adresára. Musí byť však schopný zistiť, že či užívateľom predané informácie sú v poriadku. Na tento účel sme vytvorili niekoľko prídavných tried v tomto priečinku. Kontrola dát na servery, ktorá bude rešpektovať nami navrhnutý formát pozostáva z tried takmer identických s javascriptovým modulom AIDoX, využitím len syntaxe jazyka PHP. Zadáva sa tu však nové funkcie ako je napríklad *Validate* alebo *Prepare*. *Validate* slúži na skontrolovanie správnosti údajov v dokumente ako aj jeho štruktúry a *Prepare* validný dokument prednastaví. Vytvorená knižnica teda slúži prevažne na kontrolu dát a zabezpečuje vždy správnosť údajov.

4.4 Overovanie prihlásenia a práva

Každý užívateľ aby mohol prísť do systému musí byť povinne zaregistrovaný pomocou platnej emailovej adresy. Na prihlásenie sa využíva vstavaná trieda form helper. Tá nám pomáha zachytávať chyby pri overovaní platnosti údajov. Na overenie prihlásenia sa využíva funkcia *loginValidation* ak sa užívateľ systému prihlási. nastaví sa mu práva na vykonávanie operácií, ktoré zodpovedajú jeho roly. Po prihlásení sa do *session* uloží jeho ID a premenná *isLoggedIn* je nastavená

na *true*. Aby systém zabezpečil ochranu údajov, pristupuje sa pomocou Ajax požiadaviek vždy cez jednu funkciu s danou požiadavkou a tá zabezpečí volanie požadovanej funkcie. Ak by funkcia zistila, že užívateľ už nieje viac prihlásený, pošle odpoveď, na základe ktorej bude užívateľ navigovaný na stránku prihlásenia. Podobným spôsobom postupuje aj pri overovaní práv užívateľa. ak zistí že nemá dostatočné práva, je automaticky presmerovaný na hlavnú stránku.

Záver

V závere našej práce si poďme pozrieť a zhodnotiť naše dosiahnuté výsledky. Primárnym cieľom tejto práce, bolo vytvorenie nástroja schopného definície štruktúr a šablón dokumentov, ktorým by sme boli schopný následne naviazať príslušnú logiku. Pri hľadaní výsledného riešenia sme museli však dbať na to aby nami navrhnutý systém nebol príliš náročný na obsluhu pre koncového užívateľa a bol v ňom schopný rýchlo a efektívne pracovať.

Po odskúšaní množstva neúspešných pokusov sa nám nakoniec podarilo dosiahnuť úspešného konca práce, vytvorením systému schopného tieto požiadavky implementovať. Systém je schopný rýchleho a efektívneho vytvorenia požadovaných formulárov a ich zdieľania medzi oprávnených užívateľov systému. Tí ich môžu následne vyplniť a exportovať do PDF podporovaného formátu.

Nevýhodou sa môže javiť fakt, že tvorca formuláru musí poznať aspoň základnú syntax jazyka XPath, keďže väčšina výpočtov prebieha práve v tomto jazyku. Toto by sa ako problém nejavilo ale javascript v ktorom sú výrazy vykonávané nepodporuje verziu XPath 2.0. Keďže ani nespavosť nebola dobrým nápadom pri hľadaní výsledného riešenia bola ponúknutá aspoň možnosť autoatického dopĺňania výrazov aby nebolo potrebné vypisovať celé cesty k súborom a rovnako systém sa snaží užívateľovi pomáhať kde to len je možné.

V porovnaní s ostatnými existujúcimi nástrojmi bo mohol náš formát byť rošírený do budúca elektronicú signatúru. Rovnako XPath výrazy síce nie sú nijak zložité ale niektorých užívateľov by mohli miasť, niektoré postupy vo verzií 1.0. Možné riešenie by bolo pomocou XQIB ktoré nám sprístupňuje používanie XQuery podporujúce XPath 2.0 výrazy.

I napriek tomuto je finálny produkt funkčný, a môže začať s používaním elektronických formulárov.

Použitá literatura

[1] Kosek, Jiří. *XML pro každého 2000 : Podrobný průvodce*. Vydání první. Praha : Grada. 164 s. ISBN 807-1698601.

[2] Upton, David. *CodeIgniter for rapid PHP Application Development*. 260 s. ISBN 1847191748.

[4] <https://www.formdocs.com/features.html/>

[5] <https://www.formatta.com/products/formatta-products/>

[6] <https://www.goformz.com/>

[7] <https://www.602.cz/o-spolecnosti/>

[8] <http://www.w3schools.com/js/default.asp>

[9] https://www.informatizacia.ext_dok-metodicky_pokyn_el_formulare_v10/15174c/

[2] <https://www.formulare.kosice.sk/>

[10] https://www.informatizacia.ext_dok-metodicky_pokyn_el_formulare_v10/15174c/

[2] <https://www.formulare.kosice.sk/>

Zoznam obrázkov:

| | |
|---|----|
| Obrázok 1 - FormDocs | 12 |
| Obrázok 2 - Formatta | 13 |
| Obrázok 3 - goforms..... | 13 |
| Obrázok 4 - FormFiller..... | 15 |
| Obrázok 5 - ERD primárnej časti systému | 19 |
| Obrázok 6 - Tvorca formulárov..... | 20 |
| Obrázok 7 - Hlavná stránka..... | 21 |
| Obrázok 8 - XML definícia | 23 |
| Obrázok 9 - XML element a atribúty | 24 |

Zoznam tabuliek:

| | |
|--|----|
| Tabuľka 1 - model atribúty..... | 25 |
| Tabuľka 2 - niektoré bind atribúty | 26 |